INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET AUTOMATIQUE

Accélération de l'algorithme de Newton-GMRES pour les équations de Navier-Stokes

Rémi Choquet

N° 2287

Juin 1994

PROGRAMME 6

Calcul scientifique,

modélisation

et logiciel numérique

ISSN 0249-6399

INRIA
RENNES

Accélération de l'algorithme de Newton-GMRES pour les équations de Navier-Stokes

Rémi Choquet*

Programme 6 — Calcul scientifique, modélisation et logiciel numérique
Projet ALADIN

Rapport de recherche n° 2287 — Juin 1994 — 23 pages

Résumé : On considère la résolution implicite par l'algorithme de Newton-GMRES des équations de Navier-Stokes compressibles. Dans un premier temps, on essaiera une alternative au redémarrage de l'algorithme de GMRES par le biais de préconditionnement. Puis on décrira une approche afin d'accélérer la convergence de l'algorithme de Newton consistant à réutiliser les directions de descente de l'étape précédente. A partir des résultats obtenus, on tentera de cerner le comportement et les difficultés d'une telle résolution.

Mots-clé : directions de descente, contrainte de mémoire, Navier-Stokes, Newton-GMRES, préconditionnement.

(Abstract: pto)

*choquet@irisa.fr

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 84 71 71

Newton-GMRES acceleration for the Navier-Stokes equations

Abstract: In this paper, we consider new algorithms to accelerate the matrix-free Newton-GMRES algorithm, in the context of the implicit numerical resolution of the compressible Navier-Stokes equations. First at all, we try to reduce the lost of the convergence rate of the restart inside GMRES by considering a class of preconditioners. Then, we describe a cheap way to reuse previous descent directions across Newton iterations using the Broyden update. Finally, we suggest a new algorithm and we evaluate its impact.

Key-words: descent direction, matrix-free, Navier-Stokes, Newton-GMRES, preconditioning.

1 Introduction

On s'intéresse à la résolution par des schémas implicites de problèmes de la forme

$$u_{,t} + G(u) = 0 \tag{1}$$

où $G(u) : V \rightarrow V$ est une fonction non linéaire de u. Dans le cas où (1) provient des équations de Navier-Stokes compressibles, u a $(2 + dim)$ composantes où dim désigne la dimension en espace du champ de vitesse des particules.
On considère la résolution stationnaire ou instationnaire de ces équations à l'aide d'un schéma à un pas de temps. Après des discrétisations appropriées en temps et en espace, à l'instant t_n l'équation (1) devient

$$u(n+1) + \Delta t_n H_1(u(n+1)) = \Delta t_n H_2(u(n)). \tag{2}$$

Notons que la méthode d'Euler implicite correspond à

$$H_2(u) = \frac{u}{\Delta t_n}.$$

On résout (2) par l'algorithme de Newton-GMRES [BS90]. Chaque étape de Newton linéarise le problème et le système linéaire induit est résolu par GMRES [SS86]. Il est important de souligner ici que GMRES est un solveur pour grande matrice non symétrique ne demandant que des produits matrices-vecteurs. Cette approche est utilisée [Sha88, Vui93] lorsque le problème interdit tout stockage du Jacobien en mémoire centrale. Ce cas se présente en mécanique des fluides en dimension trois d'espace mais encore lors du traitement numérique des prévisions météorologiques ou des problèmes électromagnétiques. Malheureusement, l'approche choisie est coûteuse par rapport à une résolution explicite. En particulier pour de nombreuses formulations, chaque étape de Newton conduit à la résolution d'un système non symétrique

$$J(u_i)\delta(i) = -F(u_i), \qquad i = 0, \ldots \tag{3}$$

où

$$\begin{cases} u_0 = u(n), \\ \delta(i) = u_{i+1} - u_i, \\ F(u_i) = u_i + \Delta t_n H_1(u_i) - \Delta t_n H_2(u_0), \\ J(u_i) = \frac{\partial F(u_i)}{\partial u_i} = I + \Delta t_n \frac{\partial H_1(u_i)}{\partial u}. \end{cases}$$

Souvent, la stabilité induite par la résolution implicite permet de multiplier le pas de temps par un paramètre nommé communément CFL afin d'accroître la vitesse de convergence

du schéma en temps.

La résolution par GMRES du système (3) si elle veut être efficace et peu couteuse en mémoire se heurte à ces constats.

1. Il faut un bon préconditionnement. Dans un cadre classique, le choix d'une factorisation incomplète de $J(u_i)$ comme préconditionneur couplée avec des techniques de renumérotation des nœuds du maillage donne de bons résultats [Dut91] mais ici la matrice est inconnue. Et seul le résidu fournit les informations sur le problème étudié.

2. Un redémarrage est nécessaire pour réduire le coût mémoire mais une dégradation de la convergence est alors observée par rapport à GMRES classique. Depuis peu, des algorithmes de type gradient tentent d'y apporter des réponses [VdVV92, Saa93].

3. Chaque produit jacobien-vecteur est approché par un schéma aux différences finies qui induit par là-même des erreurs pouvant dégrader la convergence superlinéaire de GMRES [VdVV93] mais aussi la qualité du vecteur de descente [CE93]

Dans cet article, deux approches ayant pour but d'accélérer l'algorithme de Newton-GMRES sont décrites. Dans le chapitre 2, on présentera l'algorithme FGMRES [Saa93] au sein duquel une classe de préconditionneur sera considérée (chapitre 3). On comparera les résultats obtenus avec GMRES plus redémarrage. Puis dans le chapitre 4, on utilise le dernier sous-espace de Krylov afin d'accélérer Newton. A l'aide de différentes expériences sur la résolution des équations de Navier-Stokes, nous en tirerons des enseignements sur le comportement de l'algorithme de résolution pour ces équations.

Notations: Soit V une matrice, on note $< V >$ le sous-espace vectoriel engendré par les vecteurs colonnes de V ; P_0^n l'ensemble des polynomes de degré n valant 1 en 0. Enfin, GMRES(n) désigne GMRES avec un redémarrage toutes les n itérations.

2 Rappels

Par la suite, on considère la résolution de tout système linéaire $J\delta = -F$ par FGMRES [Saa93] , il s'agit de l'algorithme de GMRES avec un préconditionnement à droite pouvant varier à chaque étape du processus d'Arnoldi. Voici l'algorithme utilisé tel qu'il est défini par Saad,

ALGORITHME 1: FGMRES (k_{max})
données $\varepsilon > 0$ la tolérance du critère d'arret et $\delta_0 \in \mathbb{R}^n$ l'approximation initiale,

(s1). $r_0 = -F - J\delta_0$,
$\beta = \|r_0\|_2$, $v_1 = \dfrac{r_0}{\beta}$, $k = 0$

(s2). FAIRE
$k = k + 1$
(a) $x_k = M_k^{-1} v_k$
(b) $q_{k+1} = Jx_k$, $w_{k+1} = q_{k+1} - \sum_{m=1}^{k} h_{m,k} v_m$
avec $h_{m,k} = (q_{k+1}, v_m)$, $(m = 1, \ldots, k)$
$h_{k+1,k} = \|w_{k+1}\|_2$, $v_{k+1} = \dfrac{w_{k+1}}{h_{k+1,k}}$
(c) on calcule $\rho_k = \min_{y \in \mathbb{R}^k} \|\beta e_1 - \overline{H}_k y\|_2$
JUSQU'A $\rho_k \leq \varepsilon$ ou $k = k_{max}$

(s3). $K = k$, on calcule $y_K = \text{argmin}_{y \in \mathbb{R}^K} \|\beta e_1 - \overline{H}_K y\|_2$

(s4). $\delta_K = \delta_0 + X_K y_K$

où $(\overline{H}_k)_{m,l} = \begin{cases} h_{m,l} & 1 \leq m \leq k+1, m \leq l \leq k \\ 0 & \text{sinon} \end{cases}$,
$V_k = (v_1, \ldots, v_k)$, $X_k = (x_1, \ldots, x_k)$.
La propriété essentielle de cet algorithme est

$$JX_k = V_{k+1} \overline{H}_k, \tag{4}$$

ce qui entraîne la proposition suivante.

Proposition 2.1 *La solution approchée δ_K obtenue à l'étape K de l'algorithme 1 minimise la norme $\| - F - J\delta\|_2$ sur $\delta_0 + < X_K >$.*

3 Une classe de préconditionnements

3.1 Description

L'intérêt de l'algorithme 1 réside dans sa flexibilité, on peut considérer des préconditionneurs difficiles à inverser de manière exacte.

Sachant que l'on est contraint à n'utiliser que des produits Jacobien-vecteurs, la première

idée qui vient à l'esprit est de considérer GMRES(k_{int}) comme préconditionneur. Bien sûr, la convergence de FGMRES(k_{ext}) couplé avec GMRES(k_{int}) (noté GM(k_{ext}, k_{int})) ne peut qu'approcher celle de GMRES($k_{ext} * k_{int}$) car

$$\| - F - J\delta_{k_{ext}*k_{int}}\| = \min_{p \in P_0^{k_{ext}*k_{int}}} \|p(J)r_0\|_2. \tag{5}$$

Nous allons considérer une classe plus large de préconditionneur.
Dans un article récent, Manteuffel et Otto [MO93] approximent l'opérateur

$$Au = -\triangle u + \beta_1 u_x + \beta_2 u_y + \delta u, \tag{6}$$

par

$$Bu = -\triangle u + \sigma u, \quad \sigma > 0, \tag{7}$$

qui est un opérateur symétrique défini positif plus facile à inverser que l'opérateur initial et approchant A pour σ convenablement choisi. De manière semblable, on définit ici une classe d'opérateurs à un paramètre

$$B(u_i, a) = I - \frac{\Delta t_n}{a} \frac{\partial H_1(t_n, u_i; \Delta t_n)}{\partial u_i}. \tag{8}$$

Pour $a > 1$, ceci revient à réduire le pas de temps du schéma. Outre le fait que la classe des préconditionneurs (8) contient $J(u_i) = B(u_i, 1)$, les motivations d'un tel choix sont les suivantes,

1. On constate que lorsque Δt_n décroit significativement, l'inversion de (3) est moins couteuse ainsi $B(u_i, a)$ est plus facile à inverser que $J(u_i)$ pour a assez grand (voir figure 6).

2. Une analyse similaire à celle pratiquée dans [MO93] montre un bon conditionnement de $B^{-1}(u_i, a)J(u_i)$ pour certains systèmes d'advection-diffusion à coefficient constant.

3. La construction du préconditionneur nécessite le seul résidu H_1.

Désormais, on donne une valeur limite au nombre de vecteurs pouvant être stockés en mémoire. La mise en œuvre de GM(k_{ext}, k_{int}) demande un stockage de $2 * k_{ext} + k_{int}$ vecteurs. On considère l'utilisation de (8) pour M_k^{-1} dans l'algorithme 1. Dans la prochaine section, on compare cette approche à une stratégie de redémarrage.

3.2 Résultats

On utilise la notation étendue $GM(k_{ext}, k_{int}; a)$ (ou $(k_{ext}, k_{int}; a)$ sur les graphiques) de la précédente partie pour caractériser la courbe correspondante à l'algorithme de résolution linéaire avec

- k_{ext} le nombre d'itérations externes,
- k_{int} le nombre d'itérations internes,
- et a pour l'utilisation de $B(u_i, a)$, $a \neq 1$ comme préconditionneur.

En plus des conventions précédentes, on note *epsn* la Condition de sortie de Newton et *epsg* la Condition de sortie de GMRES. Les abréviations Res_new, Res_lin et Jac_vec légendant les colonnes des tableaux, représentent respectivement les résidus relatifs de Newton et de GMRES et le nombre de produits Jacobien-vecteur.

Tout d'abord, on compare les deux algorithmes $GMRES(25)$ et $GM(10,5)$ sur la matrice provenant de [VdVV93], $A = SBS^{-1}$ avec $A, S, B \in \mathbb{R}^{100\times100}$ et $S = (1, \beta)$ une matrice bidiagonale ayant pour valeur 1 sur la diagonale et β sur la diagonale supérieure. Ce premier test est réalisé sous Matlab et l'algorithme $GM(k_{ext}, k_{int})$ est écrit à partir de la routine GMRES obtenue sous netlib [BBC*93].

Le système $Ax = b$ est résolu pour le second membre $b = (1, \ldots, 1)^t$ en prenant pour valeur initiale de x, $x_0 = 0$. Dans l'exemple considéré, B est diagonale avec des valeurs propres positives et uniformément distribuées entre 1 et 100. On pose $\beta = 0.9$, ce qui donne un conditionnement pour S de $\kappa(S) = 18.3340$. Pour cette matrice A, les résultats obtenus (figure 1) montrent la supériorité de $GM(10,5)$ sur $GM(25)$. Dans la suite de ce paragraphe, on applique cette approche au domaine étudié.

Dans ce papier, on s'intéresse au schéma d'Euler implicite en temps couplé avec le schéma d'Osher pour la décomposition du flux. On considère les simulations suivantes :

Test 1 En deux dimensions d'espace, un écoulement stationnaire à Mach 0.8, Re=5000 autour d'un Naca0012 sous une incidence nulle. On utilise le schéma d'Euler explicite en temps pour les 100 premiers pas.

Test 2 En deux dimensions d'espace, un écoulement d'Euler stationnaire à Mach 1.2 autour d'un Naca0012 sous une incidence de sept degré. On utilise le schéma d'Euler explicite en temps pour les 100 premiers pas.

Le maillage considéré possède 801 nœuds. Le code utilisé (NSC2KE) a été fourni en explicite par l'INRIA Rocquencourt. Tout les résultats présentés sont fonction du nombre de produits Jacobien-vecteur constituant le coût majeur de l'algorithme de résolution considéré. On rappele que chaque produit Jacobien-vecteur est approché par un schéma aux différences finies d'ordre un,

$$J(u).x \approx \frac{F(u+\sigma x) - F(u)}{\sigma}.$$

3.2.1 Comparaisons des $GM(k_{ext}, k_{int}; a)$

Pour des CFL de 1 et 4, on compare $GM(25)$, $GM(10,5)$ et $GM(10,5,10)$ à une étape de Newton fixée. Au travers des résultats montrés par les figures 2, 3 et 4,5, on remarque que le redémarrage reste le plus efficace pour résoudre ces systèmes pour un même encombrement mémoire.

3.2.2 Influence de $GM(k_{ext}, k_{int}; a)$ sur Newton

On considère le test 1 et on compare $GM(25)$ et $GM(10/5)$. On remarque le coût voisin des deux algorithmes de Newton.

	$k_{ext} = 25$			$k_{ext} = 10, \quad k_{int} = 5$		
i	Res_new	Res_lin	Jac_vec	Res_new	Res_lin	Jac_vec
1	0.10714719	9.7504560E-03	71	0.1071899	9.4371900E-03	90
2	2.4822087E-02	9.8797319E-03	92	1.094971E-03	9.7464239E-03	115
3	2.8761841E-04	9.9592545E-03	62	1.071891E-05	9.4649443E-03	132
4	1.6260660E-05	5.5169280E-02	125			
$CFL = 4.0, \quad epsn = 10\mathtt{E}-4, \quad \mathtt{epsg} = 10\mathtt{E}-2$						

Tableau 1 : Influence de FGMRES sur Newton.

3.2.3 Conclusion Partielle

La mise en œuvre du préconditionnement précédent a montré ses limites. En effet, lorsque le redémarrage est efficace, les alternatives proposées se sont montrées moins bonnes pour résoudre un problème linéaire. La figure 8 apporte une réponse partielle à ce relatif échec. Pour k_{int} petit, la précision obtenue lors de l'inversion de $B(u_i, a)$ reste à peu près constante lorsque a varie. Ainsi, le gain en résidu lorsque a est grand et k_{int} petit est faible. De plus, lorsque k_{int} est supérieur à k_{ext} alors le redémarrage se passe mal. On peux aussi

remarquer l'absence de convergence superlinéaire de l'algorithme de GMRES avec différences finies, ceci peut expliquer en partie la médiocre qualité de $GM(k_{ext}, k_{int})$ comparé à une stratégie de redémarrage.
Par contre, il est surprenant de constater le meilleur comportement de l'algorithme de Newton lorsque chaque système linéaire est résolu par $GM(10,5)$ au lieu de $GM(25)$. Malgré tout, on peut considérer qu'il est difficile de gagner du temps CPU par ce biais. Dans le chapitre suivant, une autre approche sera considérée dans le but d'économiser le nombre de produits Jacobien-vecteurs. On va réutiliser l'ancienne base de Krylov pour approcher l'actuelle étape de Newton. Aussi, on pourra exploiter l'autre avantage de l'utilisation d'un préconditionneur qui réside dans le fait qu'il délivre une base de vecteurs préconditionnés.

4 Réutilisation des directions de descente

Lorsque le Jacobien est constant, Vuik[Vui93] considère avec succès la projection du nouveau rédidu perpendiculairement aux anciennes directions de descente définies par l'algorithme GMRESR[VdVV93]. Ici, nous prenons en compte les variations du Jacobien en l'approchant par l'update de Broyden.

4.1 Idée générale

A l'étape $(i-1)$ de Newton, on a approché la solution de $J(u_{i-1})\delta(i-1) = -F(u_{i-1})$ par GMRES(k) ou encore par

$$\min_{y \in \mathbb{R}^k} \|r_0^{(i-1)} - J(u_{i-1}) V_k^{(i-1)} y\|_2 \tag{9}$$

où $r_0^{(i-1)} = -F(u_{i-1}) - J(u_{i-1})\delta_0^{(i-1)}$ et
$V_k^{(i-1)} = (v_1, \ldots, v_k)$ est la base d'Arnoldi de rang k.
Notons $\overline{H}_k^{(i-1)} = Q_k^{(i-1)} R_k^{(i-1)} = Q_k^{(i-1)} \begin{pmatrix} \overline{R}_k^{(i-1)} \\ 0 \end{pmatrix}$ la matrice de Hessenberg définie dans l'algorithme d'Arnoldi, supposée de rang k,

$y_k^{(i-1)}$ la solution de (9) ,
$z_k^{(i-1)} = V_k^{(i-1)} y_k^{(i-1)}$,
$\delta_k^{(i-1)} = \delta_0^{(i-1)} + z_k^{(i-1)}$ et
$r_k^{(i-1)} = -F(u_{i-1}) - J(u_{i-1})\delta_k^{(i-1)}$.

L'idée que l'on va développer à travers ce chapitre est la construction d'un préconditionneur pour $J(u_i)$ à l'aide des informations obtenues lors de la résolution de (9), plus précisement on minimise $f = F^t F$ sur le sous-espace de Krylov $V_k^{(i-1)}$,

$$c.a.d \qquad \min_{y \in \mathbb{R}^k} f(u_i + V_k^{(i-1)} y). \tag{10}$$

En remplaçant dans (10) $F(u_i + V_k^{(i-1)} y)$ par son développement limité à l'ordre un, on a

$$\min_{y \in \mathbb{R}^k} \{ f(u_i) + F(u_i)^t . J(u_i) V_k^{(i-1)} y + \frac{1}{2} (J(u_i) V_k^{(i-1)} y)^t J(u_i) V_k^{(i-1)} y \} \tag{11}$$

mais la minimisation de ce problème est coûteuse par l'évaluation du produit $J(u_i)V_k^{(i-1)}$. Par la suite, on va approcher $J(u_i)$ dans (11) par une matrice A_i fonction de $J(u_{i-1})$, puis se servir de la propriété fondamentale de GMRES liant $J(u_{i-1})$ à un problème de taille reduite

$$J(u_{i-1}) V_k^{(i-1)} = V_{k+1}^{(i-1)} \overline{H}_k^{(i-1)}.$$

Plus précisement, on approxime $J(u_i)$ par un changement à l'ordre un de $J(u_{i-1})$,

$$A_i = J(u_{i-1}) + \alpha_i t_i . s_i^t \text{ avec } s_i, t_i \in \mathbb{R}^n, \ \alpha_i \in \mathbb{R}. \tag{12}$$

4.2 Un problème approché

En particulier, on considère les deux approximations suivantes

1. $\alpha_i = 0$ ce qui revient à geler le Jacobien
2. et l'update de Broyden [DS83],

$$\delta F_i = F(u_i) - F(u_{i-1}), \tag{13}$$
$$s_i = \delta_k^{(i-1)}, \tag{14}$$
$$t_i = \delta F_i - J(u_{i-1}) s_i, \tag{15}$$
$$\alpha_i = \frac{1}{s_i^t s_i}. \tag{16}$$

On notera que A_i vérifie la propriété $A_i \delta_k^{(i-1)} = \delta F_i$ qui implique une convergence superlinéaire pour l'algorithme de Broyden sous certaines conditions de régularité. De plus, on a

$$t_i = F(u_i) + r_k^{(i-1)},$$

et

$$A_i V_k^{(i-1)} = V_{k+1}^{(i-1)} \overline{H}_k^{(i-1)} + \frac{t_i . \tilde{s_i}^t}{s_i^t s_i} \tag{17}$$

avec $\tilde{s_i} = (V_k^{(i-1)})^t s_i \in \mathbb{R}^k$.

On considère désormais la résolution du problème $J(u_i)\delta(i) = -F(u_i)$ que l'on approche par :

Pb. *Trouver* $\zeta_k^{(i)} = V_k^{(i-1)} y_k^{(i)}$ *avec* $y_k^{(i)}$ *solution de*

$$\min_{y \in \mathbb{R}^k} \| - F(u_i) - A_i V_k^{(i-1)} y \|_2 \tag{18}$$

Ce problème est équivalent à (11) en y substituant $J(u_i)$ par son approximation A_i. Notons $P_{k+1}^{(i-1)} = (V_{k+1}^{(i-1)})(V_{k+1}^{(i-1)})^t$. On distingue plusieurs cas,

- soit $\alpha_i = 0$; on peut décomposer le carré de (18) sous la forme suivante

$$\min_{y \in \mathbb{R}^k} \{ \| - (I - P_{k+1}^{(i-1)}) F(u_i) \|_2^2 + \| - P_{k+1}^{(i-1)} F(u_i) - V_{k+1}^{(i-1)} \overline{H}_k^{(i-1)} y \|_2^2 \}.$$

  Et la solution $y_k^{(i)}$ vaut

$$y_k^{(i)} = - \begin{pmatrix} (\overline{R}_k^{(i-1)})^{-1} \\ 0 \end{pmatrix} \{ (Q_k^{(i-1)})^t (V_{k+1}^{(i-1)})^t (F(u_i)) \}.$$

- soit $\alpha_i \neq 0$;

Dans un premier temps, on suppose que $t_i \notin < V_{k+1}^{(i-1)} >$. Soit $W^{(i-1)} = (V_{k+1}^{(i-1)}, \tilde{w}_{k+2})$ avec $w_{k+2} = (I - P_{k+1}^{(i-1)}) t_i$ et $\tilde{w}_{k+2} = \frac{w_{k+2}}{\|w_{k+2}\|_2}$.

Lemme 4.1 *On suppose que* $t_i \notin < V_{k+1}^{(i-1)} >$ *alors* $W^{(i-1)}$ *est une matrice orthogonale d'ordre* $k+2$ *et* $F(u_i) \in < W^{(i-1)} >$.

Preuve. Il suffit de voir que

$$F(u_i) = t_i - r_k^{(i-1)} \in t_i + < V_k^{(i-1)} > .$$

□

De plus, on a

$$V_{k+1}^{(i-1)}\overline{H}_k^{(i-1)} = W^{(i-1)} \begin{pmatrix} \overline{H}_k^{(i-1)} \\ 0 \end{pmatrix}. \tag{19}$$

En posant $\theta_i = ((t_i, v_1), \ldots, (t_i, v_{k+1}), \|w_{k+2}\|)$
$= (\theta_{i,1}, \ldots, \theta_{i,k+2})$
et

$$\tilde{Q}_k^{(i-1)} = \begin{pmatrix} Q_k^{(i-1)} & 0 \\ 0 & 1 \end{pmatrix},$$

on obtient le Lemme suivant :

Lemme 4.2 *On suppose $\alpha_i \neq 0$ et $t_i \notin < V_{k+1}^{(i-1)} >$ alors (18) est équivalent à*

$$\min_{y \in \mathbb{R}^k} \|\tilde{f}_i - \{ \begin{pmatrix} R_k^{(i-1)} \\ 0 \end{pmatrix} + \frac{\tilde{\theta}_i . \tilde{s}_i^t}{s_i^t s_i} \} y \|_2, \tag{20}$$

avec $\tilde{f}_i = -(\tilde{Q}_k^{(i-1)})^t (W^{(i-1)})^t F(u_i)$ et $\tilde{\theta}_i = (\tilde{Q}_k^{(i-1)})^t \theta_i$.

Preuve. Par définition, on a $\theta_{i,k+2}\tilde{w}_{k+2} = t_i - \sum_{j=1}^{k+1} \theta_{i,j} v_j$, ou encore $W^{(i-1)}\theta_i = t_i$. D'où,

$$V_{k+1}^{(i-1)}\overline{H}_k^{(i-1)} + \frac{t_i . \tilde{s}_i^t}{s_i^t s_i} = W^{(i-1)} \{ \begin{pmatrix} \overline{H}_k^{(i-1)} \\ 0 \end{pmatrix} + \frac{\theta_i . \tilde{s}_i^t}{s_i^t s_i} \}.$$

Ainsi, on a une suite de problèmes équivalents, par (17), (19), le lemme 4.1, l'orthogonalité de $\tilde{Q}_k^{(i-1)}$,

$$\begin{aligned}
(18) &\iff \min_{y \in \mathbb{R}^k} \|W^{(i-1)} \left((W^{(i-1)})^t F_i - \{ \begin{pmatrix} \overline{H}_k^{(i-1)} \\ 0 \end{pmatrix} + \frac{\theta_i . \tilde{s}_i^t}{s_i^t s_i} \} y \right) \|_2 \\
&\iff \min_{y \in \mathbb{R}^k} \|\tilde{Q}_k^{(i-1)} \left(\tilde{f}_i - \{ \begin{pmatrix} R_k^{(i-1)} \\ 0 \end{pmatrix} + \frac{\tilde{\theta}_i . \tilde{s}_i^t}{s_i^t s_i} \} y \right) \|_2 \\
&\iff \min_{y \in \mathbb{R}^k} \|\tilde{f}_i - \{ \begin{pmatrix} R_k^{(i-1)} \\ 0 \end{pmatrix} + \frac{\tilde{\theta}_i . \tilde{s}_i^t}{s_i^t s_i} \} y \|_2
\end{aligned}$$

□

Remarque 4.1 *On a $\tilde{f}_i = \tilde{\theta}_i + \beta q_{1,k+1}^{(i-1)} e_{k+1}$, avec $(e_{k+1})_{,l} = \begin{cases} 1 \ si \ l = k+1 \\ 0 \ sinon \end{cases}$.*

Désormais, on suppose que $t_i \in < V_{k+1}^{(i-1)} >$. On obtient un résultat similaire au lemme 4.2.

Lemme 4.3 *On suppose $\alpha_i \neq 0$ et $t_i \in < V_{k+1}^{(i-1)} >$ alors (18) est équivalent à*

$$\min_{y \in \mathbb{R}^k} \|\tilde{f}_i - \{ \begin{pmatrix} R_k^{(i-1)} \\ 0 \end{pmatrix} + \frac{\tilde{\theta}_i . \tilde{s}_i^t}{s_i^t s_i} \} y\|_2, \tag{21}$$

avec $\tilde{f}_i = -(Q_k^{(i-1)})^t (V_{k+1}^{(i-1)})^t F_i$ et $\tilde{\theta}_i = (Q_k^{(i-1)})^t (V_{k+1}^{(i-1)})^t t_i$.

Preuve. Elle est semblable à celle du lemme 4.2 □

Par cette approche, la résolution de (18) s'effectue de manière peu couteuse. En effet (voir [GVL89]), $\exists X \in \mathbb{R}^{(k+2)\times(k+2)}$ orthogonale tel que $X\tilde{\theta}_i = \pm\|\tilde{\theta}_i\|_2 e_1$, ainsi

$$(20, 21) \Longleftrightarrow \min_{y \in \mathbb{R}^k} \|X\tilde{f}_i - L_k^{(i)} y\|_2, \tag{22}$$

avec $L_k^{(i)} = X \begin{pmatrix} R_k^{(i-1)} \\ 0 \end{pmatrix} - \pm \frac{\|\tilde{\theta}_i\|_2 e_1 \tilde{s}_i^t}{s_i^t s_i}$ étant une matrice de Hessenberg supérieure. Il suffit dès lors de rendre la matrice $L_k^{(i)}$ rectangulaire à l'aide de rotations de Givens.

4.3 Algorithme

L'utilisation du problème (18) pour approcher (10) ne convient pas si F est localement linéaire autour de u_{i-1}, en effet :

Proposition 4.1 *Si F vérifie $F(u_i) = F(u_{i-1}) + J(u_{i-1})\delta_k^{(i-1)}$ alors la solution de (18) est triviale.*

Preuve. Par hypothèse, $t_i = 0$ et la propriété de minimisation du résidu de GMRES donne $F(u_i) = F(u_{i-1}) + J(u_{i-1})\delta_k^{(i-1)} \perp J(u_{i-1})V_k^{(i-1)}$. □

On suppose désormais que $F(u_i) \neq F(u_{i-1}) + J(u_{i-1})\delta_k^{(i-1)}$, aussi on met en œuvre l'algorithme suivant, au sein duquel, après chaque étape de Newton (**s1**), une étape de l'accélération est considérée (**s2**).

ALGORITHME 2: Newton-GMRES accéléré

$0 < \eta < 1$ et u_0 donnés, $i = -1$

FAIRE

$i = i + 1$

(s1). Résoudre $J(u_i)\delta(i) = -F(u_i)$ par exemple par GMRES(k_{ext}).
On a $V_K^{(i)}, Q_K^{(i)}, \overline{R}_K^{(i)}$ et $\delta_K^{(i)}$ la solution approchée.
$\tilde{u}_{i+1} = u_i + \delta_K^{(i)}$.
(s2). Résoudre (18) par l'algorithme 3 (dans le cas où $\alpha_{i+1} \neq 0$, il est défini dans l'annexe A) et on pose :
ω=Accélération($\|F(u_i)\|_2, F(\tilde{u}_{i+1}), \delta_K^{(i)}, V_K^{(i)}, \overline{R}_K^{(i)}, Q_K^{(i)}$).
Si $\omega \leq \eta$ alors $u_{i+1} = \tilde{u}_{i+1} + \zeta_K^{(i+1)}$.
JUSQU'A *convergence*

Le critêre η reste à définir, on peut montrer que lorsqu'il est suffisament petit alors $\zeta_K^{(i+1)}$ est une direction de descente pour f en $\tilde{u}_{i+1}$.

Proposition 4.2 *On suppose que $J(u_i)$ est lipschitzienne de constante γ sur D_i un ouvert convexe contenant $B(u_i, r_i) \subset D_i$ et $\|\delta_K^{(i)}\|_2 \leq r_i$. Si η vérifie :*

$$\|F(\tilde{u}_{i+1})\|_2(\eta - 1) + \frac{3\gamma}{2}\|\zeta_K^{(i+1)}\|_2\|\delta_K^{(i)}\|_2 < 0$$

alors $\zeta_K^{(i+1)}$ est une direction de descente pour f en $\tilde{u}_{i+1}$.

Preuve. De par le lemme 8.2.1 (p 175) [DS83], on a

$$\|J(\tilde{u}_{i+1}) - A_{i+1}\|_2 \leq \frac{3\gamma}{2}\|\delta_K^{(i)}\|_2.$$

Ainsi,

$$\begin{aligned}
F^t(\tilde{u}_{i+1})J(\tilde{u}_{i+1})\zeta_K^{(i+1)} &= F^t(\tilde{u}_{i+1})A_{i+1}\zeta_K^{(i+1)} + F^t(\tilde{u}_{i+1})(J(\tilde{u}_{i+1}) - A_{i+1})\zeta_K^{(i+1)} \\
\|F^t(\tilde{u}_{i+1})J(\tilde{u}_{i+1})\zeta_K^{(i+1)}\|_2 &\leq \|F(\tilde{u}_{i+1})\|_2(-\|F(\tilde{u}_{i+1})\|_2 + \| - F(\tilde{u}_{i+1}) - A_{i+1}\zeta_K^{(i+1)}\|_2) \\
&\quad + \|F(\tilde{u}_{i+1})\|_2\|J(\tilde{u}_{i+1}) - A_{i+1}\|_2\|\zeta_K^{(i+1)}\|_2 \\
&\leq \|F(\tilde{u}_{i+1})\|_2^2(\eta - 1) + \frac{3\gamma}{2}\|F(\tilde{u}_{i+1})\|_2\|\zeta_K^{(i+1)}\|_2\|\delta_K^{(i)}\|_2.
\end{aligned}$$

□

Remarque 4.2 *Une extension de l'algorithme 2 peut être faite en considérant plusieurs étapes de l'accélération (**s2**). Elle est directe en laissant le jacobien constant. Elle demande des mises à jour successives de (22) dans l'autre cas.*

4.4 Extension au préconditionnement

On se place ici dans le cadre de FGMRES, on a résolu à l'étape $(i-1)$ de Newton

$$\min_{y\in\mathbb{R}^k} \|r_0^{(i-1)} - (J(u_{i-1})M^{-1})V_k^{(i-1)}y\|_2. \tag{23}$$

Le problème approché (18) devient :

Trouver $\zeta_k^{(i)} = X_k^{(i-1)}y_k^{(i)}$ avec $y_k^{(i)}$ solution de

$$\min_{y\in\mathbb{R}^k} \| - F(u_i) - (A_iM^{-1})V_k^{(i-1)}y\|_2 \tag{24}$$

et A_iM^{-1} l'update de Broyden de $J(u_{i-1})M^{-1}$ défini par

$$A_iM^{-1} = J(u_{i-1})M^{-1} + \frac{t_i.s_i^t}{\|s_i\|_2^2}, \tag{25}$$

où $\begin{cases} \delta F_i = F(u_i) - F(u_{i-1}), \\ M^{-1}s_i = \delta_k^{(i-1)} \Longleftrightarrow s_i = M\delta_0^{(i-1)} + V_k^{(i-1)}y_k^{(i)}, \\ t_i = \delta F_i - J(u_{i-1})M^{-1}s_i. \end{cases}$

Remarque 4.3 *L'évaluation de $M\delta_0^{(i-1)}$ n'est pas toujours possible sauf pour le cas particulier ou $\delta_0^{(i-1)} = 0$ cependant on peut considérer à la place $J\delta_0^{(i-1)}$.*

4.5 Expériences

Dans cette section, on expérimente l'algorithme 2 en prenant consécutivement $GM(k_{ext})$ (partie 4.5.1) puis $GM(k_{ext}, k_{int})$ (partie 4.5.2) comme solveur linéaire des étapes de Newton (**s1**). Notons que pour $GM(k_{ext})$, le sous espace de Krylov considéré est défini lors de l'ultime redémarrage. Par contre dans le cas de GM(k_{ext},k_{int}), on s'interdit tout redémarrage.

4.5.1 Avec $GM(k_{ext})$

Les résultats montrent le bon comportement de Newton accéléré pour le Test subsonique (tables 3 et 4). Par contre, seul un gain minime a été acquis lors de la troisième itération de Newton pour le test supersonique (table 2). Dans ce cas, il est difficile d'évaluer le réel apport de l'accélération, car on a constaté une perte de précision du schéma aux différences finies approximant le produit Jacobien-vecteur, qui engendre une stagnation du

résidu de Newton.

i	Rés_new	Rés_acc	ω
1	5.2937022E-03	-	0.9828690
2	7.8534502E-05	-	0.5436610
2a		5.1588062E-05	
3	2.7971852E-06	1.9758955E-06	0.8445914
4	1.8958609E-06	1.9325031E-06	0.4385553
4a		1.9063627E-06	
$CFL = 1.0,\ k_{ext} = 25,\ epsn = 1.\mathtt{E}-6,\ \mathtt{epsg} = 1.\mathtt{E}-3$			

Tableau 2 : Test 2 ; Accélération.

i	Rés_new	Rés_acc	ω
1	1.5136439E-02	-	0.987992
2	5.7444049E-04	-	0.263404
2a		1.5131066E-04	
3	6.1081063E-06	2.8264898E-06	
$CFL = 1.0,\ k_{ext} = 25,\ epsn = 1.\mathtt{E}-4,\ \mathtt{epsg} = 1.\mathtt{E}-2$			

Tableau 3 : Test 1 ; Accélération.

i	Rés_new	Rés_acc	ω
1	0.10714719	-	0.9760904
2	2.4822087E-02	-	0.04694258
2a		3.4642479E-03	
3	2.8761841E-04	8.0443732E-05	
4	1.6260660E-05		
$CFL = 4.0,\ k_{ext} = 25,\ epsn = 1.\mathtt{E}-4,\ \mathtt{epsg} = 1.\mathtt{E}-2$			

Tableau 4 : Test 1 ; Accélération.

4.5.2 Avec $GM(k_{ext}, k_{int})$

Ce test (table 5) montre malgré un nombre d'étapes de Newton plus grand, un gain sur le nombre de produits Jacobien-vecteurs.

i	Rés_new	Rés_acc	ω
1	0.1089500	-	0.9985143
2	2.6096256E-03	-	0.9999719
3	6.6151229E-02	-	0.0222083
3a		2.6036835E-02	
4	9.4885447E-04	7.3040641E-04	0.8510803
5	1.3515879E-04	1.5681853E-02	0.0299612
5a		4.4366440E-02	
6	2.0036848E-05	3.1411471E-03	0.6709428
6a		2.1338247E-03	
7		5.5520138E-05	
Jac_vec	450	400	

$CFL = 4.0, \quad k_{ext} = 10, k_{int} = 10, \quad epsn = 1.\mathtt{E}-4, \quad \mathtt{epsg} = 3.\mathtt{E}-2$

Tableau 5 : Test 1 ; Accélération.

5 Conclusion

L'étude du préconditionnement proposé dans le chapitre 3 n'a pas montré clairement d'avantages sur le redémarrage. Il faudrait reconsidérer l'étude lorsque la perte de convergence au redémarrage est significative. Par contre, l'algorithme de Newton accéléré a montré des résultats encourageants. Il serait souhaitable d'étendre les investigations à d'autres problèmes. Dans le futur, on considérera le même type d'algorithme pour des δ-schémas.

Annexe A :Algorithme d'accélération et Complexité

ALGORITHME 3: Accélération(β, f, s, V_{k+1}, R_{k+1}, Q_{k+1})
$\beta \in \mathbb{R}$; $f \in \mathbb{R}^n$; $s \in \mathbb{R}^n$; $V_{k+1} \in \mathbb{R}^{(n+1)\times k}$; $R_{k+1} \in \mathbb{R}^{k\times k}$;
$Q_{k+1} \in \mathbb{R}^{(k+1)\times(k+1)}$

(s1) {Constitution du problème approché (20) ..... }
$x = \beta\ q_{1,k+1}$
$r = x\ V_{k+1}\ q_{k+1}$
$t = f + r$
$w_m = v_m$ pour $m = 1, \ldots, k+1$
$w_{k+2} = (I - V_{k+1}.V_{k+1}^t)t$
$\theta = ((t, v_1), \ldots, (t, v_{k+1}), \|w_{k+2}\|_2)$

$w_{k+2} = \frac{w_{k+2}}{\|w_{k+2}\|_2}$

$\tilde{\theta} = \begin{pmatrix} Q_k & 0 \\ 0 & 1 \end{pmatrix}^t \theta$

$\tilde{f} = -\tilde{\theta} + x * (0, \ldots, 0, 1, 0)^t$

$\tilde{s} = V_{k+1}^t s$

(s2) {Constitution du problème approché (22) ..... }

Calcul de $X \in \mathbb{R}^{(k+2)\times(k+2)}$ orthogonale tel que $X\tilde{\theta}_i = \pm\|\tilde{\theta}_i\|_2 e_1$

$\tilde{f} = X\tilde{f}$

$L = X R_{k+1} - \pm\frac{\|\tilde{\theta}\|_2 e_1 \tilde{s}^t}{s^t s}$

(s3) {Triangularisation de L matrice de Hessenberg supérieure . }

Fact. QR de $L = ST$ avec S orthogonale et T triangulaire supérieure de rang k

$\tilde{f} = S^t\tilde{f}$

(s4) {Minimisation du problème $\|\tilde{f} - Ty\|_2$ ..... }

$y = T^{-1}(\tilde{f}_1, \ldots, \tilde{f}_k)^t$

Rendre $\tilde{f}_{k+1}^2 + \tilde{f}_{k+2}^2$

étape	Saxpy	Sdot	Sscal	Flops
(s1)	$2k+3$	$2k+3$	$k+2$	$6(k+1)$
(s2)+(s3)				$13(k+2)^2 + 12(k+2)$
(s4)				$\frac{k^2}{2}$
Total	$2k+3$	$2k+3$	$k+2$	$13(k+2)^2 + O(k)$

Tableau 6 : Complexité de l'algorithme d'accélération.

Annexe B :Courbes

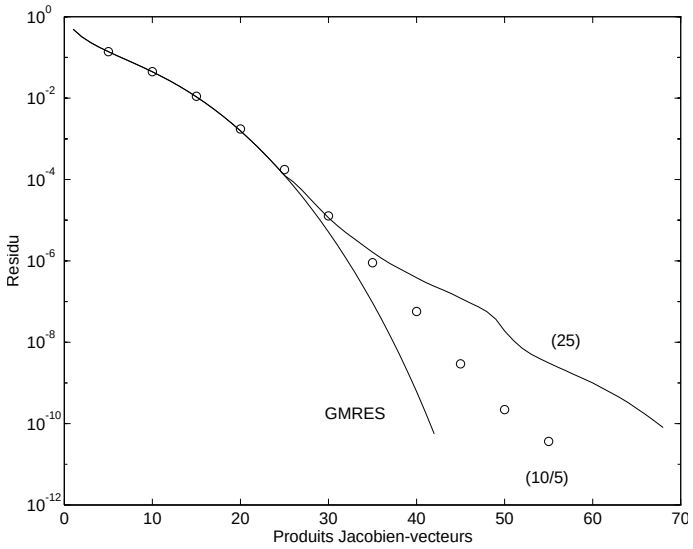

Figure 1 : $A = SBS^{-1}$, $\beta = 0.9$

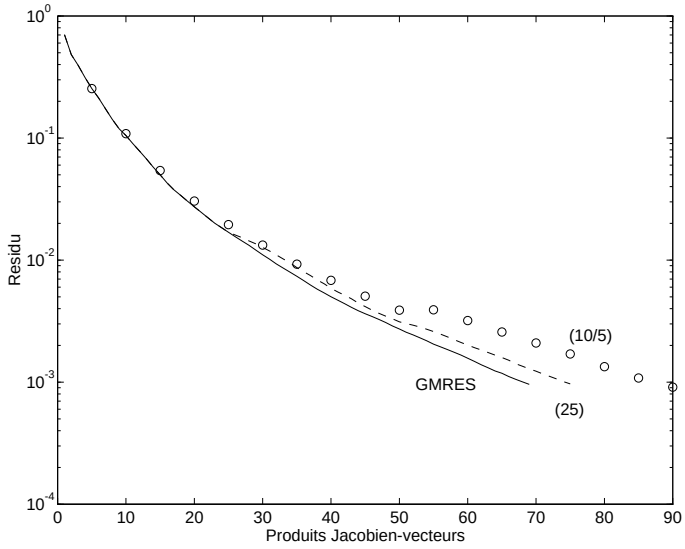

Figure 2 : Test1, CFL=1.0

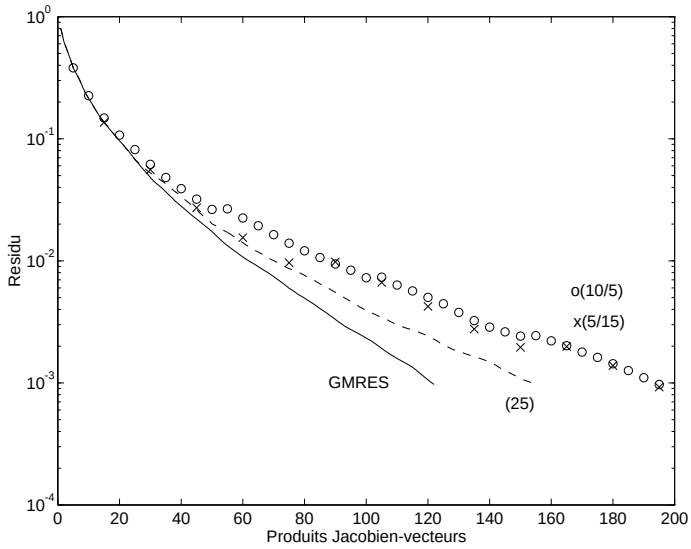

Figure 3 : Test1, CFL=4.0

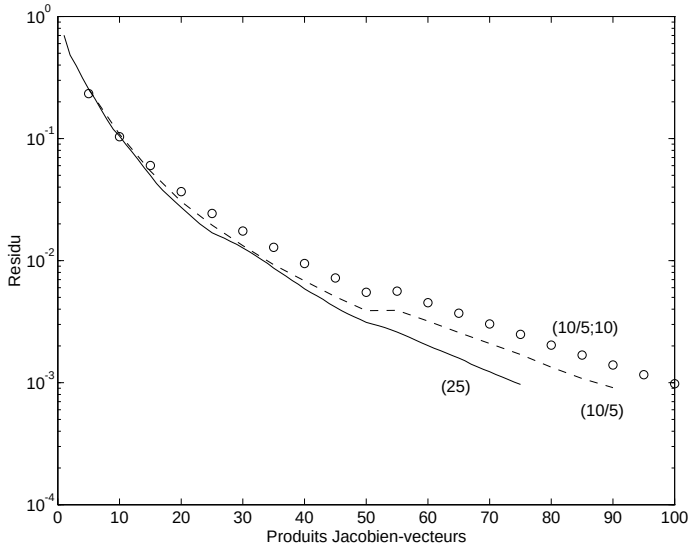

Figure 4 : Test1, CFL=1.0, a=10

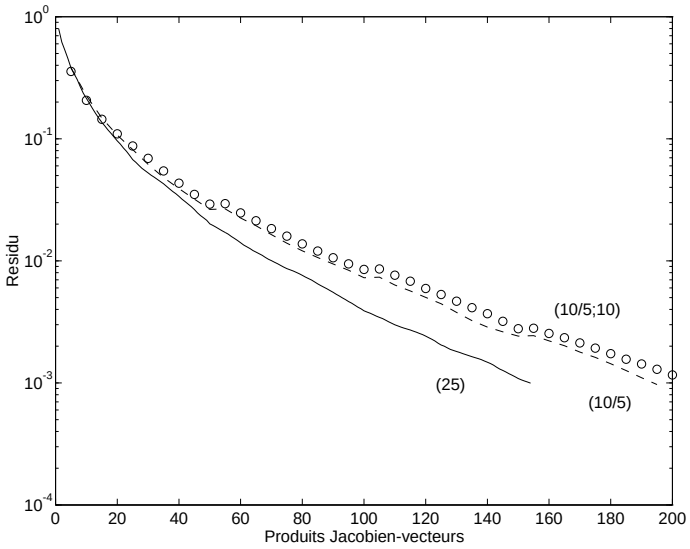

Figure 5 : Test1, CFL=4.0, a=10

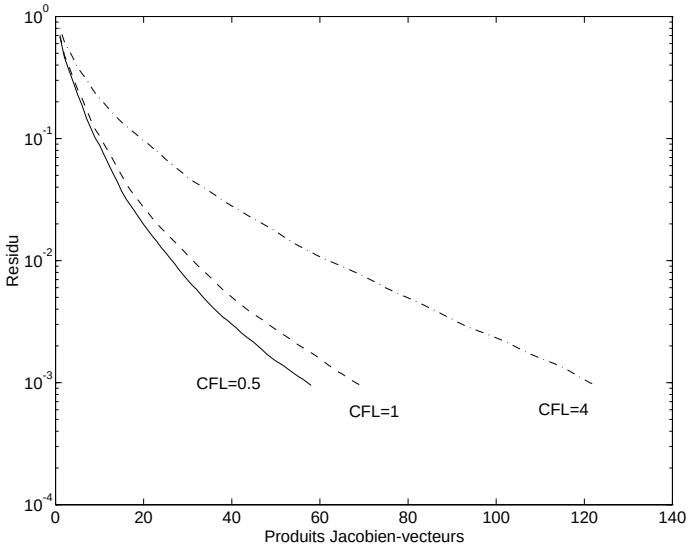

Figure 6 : Test1, comparaison pour différents CFL

References

[BBC*93] R. Barret, M. Berry, T. Chan, J. Demmel, J. Donat, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the solution of linear systems: buiding blocks for iterative methods.* SIAM / netlib, 1993.

[BS90] P.N Brown and Y Saad. Hybrid krylov methods for nonlinear systems of equations. *SIAM J.Sci.Stat.Comput*, 11(3):450–481, Mai 1990.

[CE93] R Choquet and J Erhel. Some convergence results for the newton-gmres algorithm. Research Report No 755, IRISA, Septembre 1993.

[DS83] J.E Dennis and R.B Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall series in Computational Mathematics, 1983.

[Dut91] L.C Dutto. The effect of ordering on preconditioned gmres algorithm applied to solve the compressible navier-stokes equations on unstructured grids. July 1991. Submitted to Int. J. for Numerical Methods in Engineering.

[GVL89] G.H Golub and C.F Van Loan. *Matrix Computations. second edition.* John Hopkins, 1989.

[MO93] T. Manteuffel and J. Otto. Optimal equivalent preconditioners. *SIAM J.Numer.Anal*, 30:790–812, June 1993.

[Saa93] Y Saad. A flexible inner-outer preconditioned gmres algorithm. *Siam J. Sci. Comput.*, 14(2):461–469, March 1993.

[Sha88] F Shakib. *Finite Element Analysis of the compressible Euler and Navier-Stokes equations.* PhD thesis, Stanford University, 1988.

[SS86] Y Saad and H Schultz. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[VdVV92] H.A. Van der Vorst and C. Vuik. *GMRESR: A family of nested GMRES Methods.* PREPRINT 749, Utrecht University, Department of Mathematics, October 92.

[VdVV93] H.A. Van der Vorst and C. Vuik. The superlinear convergence behaviour of gmres. *Journal of Computational and Applied Mathematics*, (48):327–341, 1993.

[Vui93] C Vuik. *Fast iterative solvers for the discretized incompressible Navier-Stokes equations.* Technical Report 93-98, Delft University of Technology, 1993.

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399